# TASK 15-Social Media Engagement

**Description:** Platform collects likes, comments, shares, post type, user demographics, and timestamps. Managers want to optimize content strategy.

## Dataset:

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Post_ID | Timestam | Likes | Comment | Shares | Post_Type | User_Dem | User_Loca | Latitude | Longitude | Post_Text | Comment | Engageme | Day_Type |
| 2 | 1001 | ######## | 520 | 45 | 12 | Video | Gen Z | New York | 40.71 | -74.01 | "Thrilled v | U-111 | 0.06 | Weekend |
| 3 | 1002 | ######## | 150 | 10 | 3 | Image | Millennia | London | 51.51 | 0.13 | "Throwba | U-789 | 0.03 | Weekday |
| 4 | 1003 | ######## | 950 | 120 | 50 | Carousel | Gen Z | Mumbai | 19.08 | 72.88 | "Top 5 tip | U-111 | 0.1 | Weekday |
| 5 | 1004 | ######## | 80 | 5 | 1 | Text | Boomer | New York | 40.71 | -74.01 | "A quick t | U-456 | 0.01 | Weekday |
| 6 | 1005 | ######## | 1200 | 180 | 95 | Video | Gen Z | London | 51.51 | 0.13 | "Must-wa | U-222 | 0.18 | Weekend |
| 7 | 1006 | ######## | 300 | 20 | 5 | Image | Millennia | Mumbai | 19.08 | 72.88 | "Weeken | U-789 | 0.045 | Weekday |
| 8 | 1007 | ######## | 210 | 15 | 2 | Text | Boomer | New York | 40.71 | -74.01 | "Just finis | U-111 | 0.025 | Weekday |
| 9 | 1008 | ######## | 1500 | 250 | 150 | Video | Gen Z | London | 51.51 | 0.13 | "This filte | U-456 | 0.22 | Weekday |
| 10 | 1009 | ######## | 600 | 70 | 20 | Carousel | Millennia | New York | 40.71 | -74.01 | "Swipe to | U-222 | 0.085 | Weekend |
| 11 | 1010 | ######## | 110 | 8 | 1 | Image | Boomer | Mumbai | 19.08 | 72.88 | "Simple p | U-789 | 0.015 | Weekend |
| 12 | 1011 | ######## | 45 | 3 | 0 | Text | Gen Z | New York | 40.71 | -74.01 | "Morning | U-456 | 0.005 | Weekday |
| 13 | 1012 | ######## | 800 | 90 | 35 | Video | Millennia | London | 51.51 | 0.13 | "Behind tl | U-111 | 0.11 | Weekday |
| 14 | 1013 | ######## | 400 | 30 | 8 | Image | Boomer | Mumbai | 19.08 | 72.88 | "New reci | U-789 | 0.05 | Weekday |
| 15 | 1014 | ######## | 1800 | 300 | 200 | Carousel | Gen Z | New York | 40.71 | -74.01 | "The ultin | U-222 | 0.3 | Weekday |
| 16 | 1015 | ######## | 90 | 6 | 1 | Text | Millennia | London | 51.51 | 0.13 | "Just thinl | U-456 | 0.012 | Weekday |
| 17 | 1016 | ######## | 750 | 85 | 40 | Video | Boomer | Mumbai | 19.08 | 72.88 | "Flashbac | U-111 | 0.095 | Weekend |
| 18 | 1017 | ######## | 250 | 18 | 4 | Image | Gen Z | New York | 40.71 | -74.01 | "My morn | U-789 | 0.035 | Weekend |
| 19 | 1018 | ######## | 120 | 9 | 2 | Text | Millennia | London | 51.51 | 0.13 | "Feeling i | U-456 | 0.018 | Weekday |
| 20 | 1019 | ######## | 1000 | 150 | 70 | Carousel | Boomer | Mumbai | 19.08 | 72.88 | "See what | U-222 | 0.15 | Weekday |
| 21 | 1020 | ######## | 60 | 4 | 0 | Image | Gen Z | New York | 40.71 | -74.01 | "Late nigh | U-789 | 0.008 | Weekday |
| 22 | 1021 | ######## | 190 | 14 | 3 | Text | Millennia | London | 51.51 | 0.13 | "Thoughts | U-111 | 0.022 | Weekday |
| 23 | 1022 | ######## | 350 | 25 | 6 | Video | Boomer | Mumbai | 19.08 | 72.88 | "Quick wc | U-456 | 0.04 | Weekday |
| 24 | 1023 | ######## | 2000 | 400 | 300 | Carousel | Gen Z | New York | 40.71 | -74.01 | "Viral cha | U-222 | 0.4 | Weekend |
| 25 | 1024 | ######## | 130 | 7 | 1 | Image | Millennia | London | 51.51 | 0.13 | "Coffee b | U-789 | 0.016 | Weekend |
| 26 | 1025 | ######## | 500 | 38 | 10 | Video | Boomer | New York | 40.71 | -74.01 | "Gardenir | U-111 | 0.055 | Weekday |
| 27 | 1026 | ######## | 2200 | 500 | 450 | Video | Gen Z | Mumbai | 19.08 | 72.88 | "Mind-blc | U-456 | 0.5 | Weekday |
| 28 | 1027 | ######## | 70 | 5 | 0 | Text | Millennia | London | 51.51 | 0.13 | "Happy ne | U-789 | 0.009 | Weekday |
| 29 | 1028 | ######## | 100 | 6 | 1 | Image | Boomer | New York | 40.71 | -74.01 | "Sunset vi | U-222 | 0.014 | Weekday |
| 30 | 1029 | ######## | 1400 | 200 | 110 | Carousel | Gen Z | Mumbai | 19.08 | 72.88 | "Before ai | U-111 | 0.16 | Weekday |
| 31 | 1030 | ######## | 50 | 2 | 0 | Text | Millennia | London | 51.51 | 0.13 | "Monday | U-789 | 0.006 | Weekend |
| 32 | | | | | | | | | | | | | | |

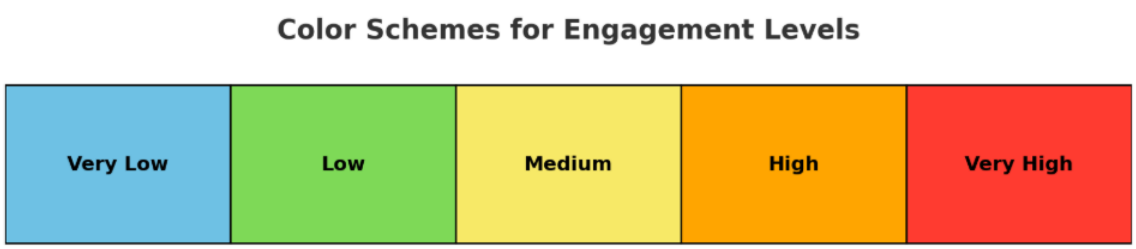
## Questions:

1. **Explain color schemes for engagement levels.**
   **Code**:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
import numpy as np
import re
from wordcloud import WordCloud
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import KMeans
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
import squarify
import folium
from folium.plugins import MarkerCluster
import networkx as nx
```

**Visualization:**




1. **Inference:** Color Gradient Meaning:
   The visualization follows a sequential gradient from cool (blue) to warm (red) tones, symbolizing increasing levels of engagement intensity.
2. Interpretation by Level:
   - Very Low: Represents minimal or negligible user engagement.
   - Low: Slight interaction, but below desired activity.

- □ Medium: Average or expected engagement levels.
- □ High: Strong audience activity and content performance.
- Very High: Peak interaction — viral or highly engaging content.

3. Visual Insight:
   The design makes it easy to identify engagement strength at a glance — warmer colors (orange/red) instantly draw attention to high-performing segments, while cooler shades (blue/green) indicate areas needing improvement.

---

2. **Design a visualization pipeline from social media data to dashboard.**
   **Code:**
3. data = {
4.     'Post_ID': [1001,1002,1003,1004,1005,1006,1007,1008,1009,1010,1011,1012,1013,1014,1015,1016,1017,1018,1019,1020,1021,1022,1023,1024,1025,1026,1027,1028,1029,1030],
5.     'Timestamp': ['2025-10-26 18:00:00','2025-10-23 10:30:00','2025-10-25 14:00:00','2025-10-27 09:00:00','2025-10-26 21:00:00','2025-10-24 16:30:00','2025-10-22 11:00:00','2025-10-20

19:45:00','2025-10-19 12:00:00','2025-10-18 15:30:00','2025-10-17 08:00:00','2025-10-16 20:00:00','2025-10-15 13:00:00','2025-10-14 17:00:00','2025-10-13 14:30:00','2025-10-12 16:00:00','2025-10-11 11:30:00','2025-10-10 19:00:00','2025-10-09 09:30:00','2025-08-08 22:00:00','2025-08-07 13:45:00','2025-08-06 17:30:00','2025-08-05 10:00:00','2025-08-04 15:00:00','2025-08-03 11:00:00','2025-08-02 18:30:00','2025-08-01 12:45:00','2025-09-30 20:30:00','2025-09-29 14:15:00','2025-09-28 16:45:00'],

6. 'Likes': [520,150,950,80,1200,300,210,1500,600,110,45,800,400,1800,90,750,250,120,1000,60,190,350,2000,130,500,2200,70,100,1400,50],
7. 'Comments': [45,10,120,5,180,20,15,250,70,8,3,90,30,300,6,85,18,9,150,4,14,25,400,7,38,500,5,6,200,2],
8. 'Shares': [12,3,50,1,95,5,2,150,20,1,0,35,8,200,1,40,4,2,70,0,3,6,300,1,10,450,0,1,110,0],
9. 'Post_Type': ['Video','Image','Carousel','Text','Video','Image','Text','Video','Carousel','Image','Text','Video','Image','Carousel','Text','Video','Image','Text','Carousel','Image','Text','Video','Carousel','Image','Video','Video','Text','Image','Carousel','Text'],
10. 'User_Demographic': ['Gen Z','Millennial','Gen Z','Boomer','Gen Z','Millennial','Boomer','Gen Z','Millennial','Boomer','Gen Z','Millennial','Boomer','Gen Z','Millennial','Boomer','Gen Z','Millennial','Boomer','Gen

Z','Millennial','Boomer','Gen Z','Millennial','Boomer','Gen Z','Millennial','Boomer','Gen Z','Millennial'],

11. 'User_Location': ['New York','London','Mumbai','New York','London','Mumbai','New York','London','New York','Mumbai','New York','London','Mumbai','New York','London','Mumbai','New York','London','Mumbai','New York','London','Mumbai','New York','London','New York','Mumbai','London','New York','Mumbai','London'],
12. 'Latitude': [40.71,51.51,19.08,40.71,51.51,19.08,40.71,51.51,40.71,19.08,40.71,51.51,19.08,40.71,51.51,19.08,40.71,51.51,19.08,40.71,51.51,19.08,40.71,51.51,40.71,19.08,51.51,40.71,19.08,51.51],
13. 'Longitude': [-74.01,0.13,72.88,-74.01,0.13,72.88,-74.01,0.13,-74.01,72.88,-74.01,0.13,72.88,-74.01,0.13,72.88,-74.01,0.13,72.88,-74.01,0.13,72.88,-74.01,0.13,-74.01,72.88,0.13,-74.01,72.88,0.13],
14. 'Post_Text': ["Thrilled with the outcome! #creative #design","Throwback Thursday to the best trip. #travel","Top 5 tips for success this week! #hustle","A quick thought on current events.","Must-watch tutorial! □ #tutorial #video","Weekend vibes starting early. #fun","Just finished a great book.","This filter is everything! 😂 #funny","Swipe to see the transformation!","Simple pleasures today.","Morning thoughts... ☕","Behind the scenes look. #makingof","New recipe alert! 😋","The ultimate guide is here! #guide #pro","Just thinking out loud.","Flashback to

summer. ☀ #vacation","My morning routine.","Feeling inspired.","See what we built! #innovation #tech","Late night snack.","Thoughts on the latest news?","Quick workout motivation.","Viral challenge accepted! 🔥 #challenge","Coffee break perfection.","Gardening tips for beginners.","Mind-blowing effect! □ #magic #viral","Happy new month!","Sunset view from the office.","Before and after reveal! #makeover","Monday mood."],

15. 'Commenter_ID': ['U-111','U-789','U-111','U-456','U-222','U-789','U-111','U-456','U-222','U-789','U-456','U-111','U-789','U-222','U-456','U-111','U-789','U-456','U-222','U-789','U-111','U-456','U-222','U-789','U-111','U-456','U-789','U-222','U-111','U-789'],

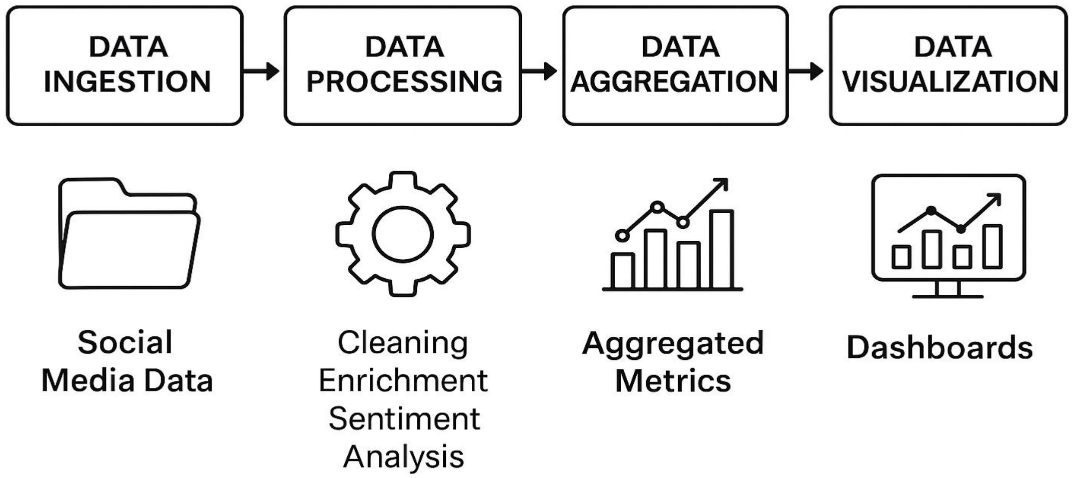
16. 'Engagement_Score': [0.060,0.030,0.100,0.010,0.180,0.045,0.025,0.220,0.085,0.015,0.005,0.110,0.050,0.300,0.012,0.095,0.035,0.018,0.150,0.008,0.022,0.040,0.400,0.016,0.055,0.500,0.009,0.014,0.160,0.006],

17. 'Day_Type': ['Weekend','Weekday','Weekday','Weekday','Weekend','Weekday','Weekday','Weekday','Weekend','Weekend','Weekday','Weekday','Weekday','Weekday','Weekday','Weekend','Weekend','Weekday','Weekday','Weekday','Weekday','Weekday','Weekend','Weekend','Weekday','Weekday','Weekday','Weekday','Weekday','Weekend']

18. }

19. df = pd.DataFrame(data)

**Visualization:**




## 🔍 Inference:

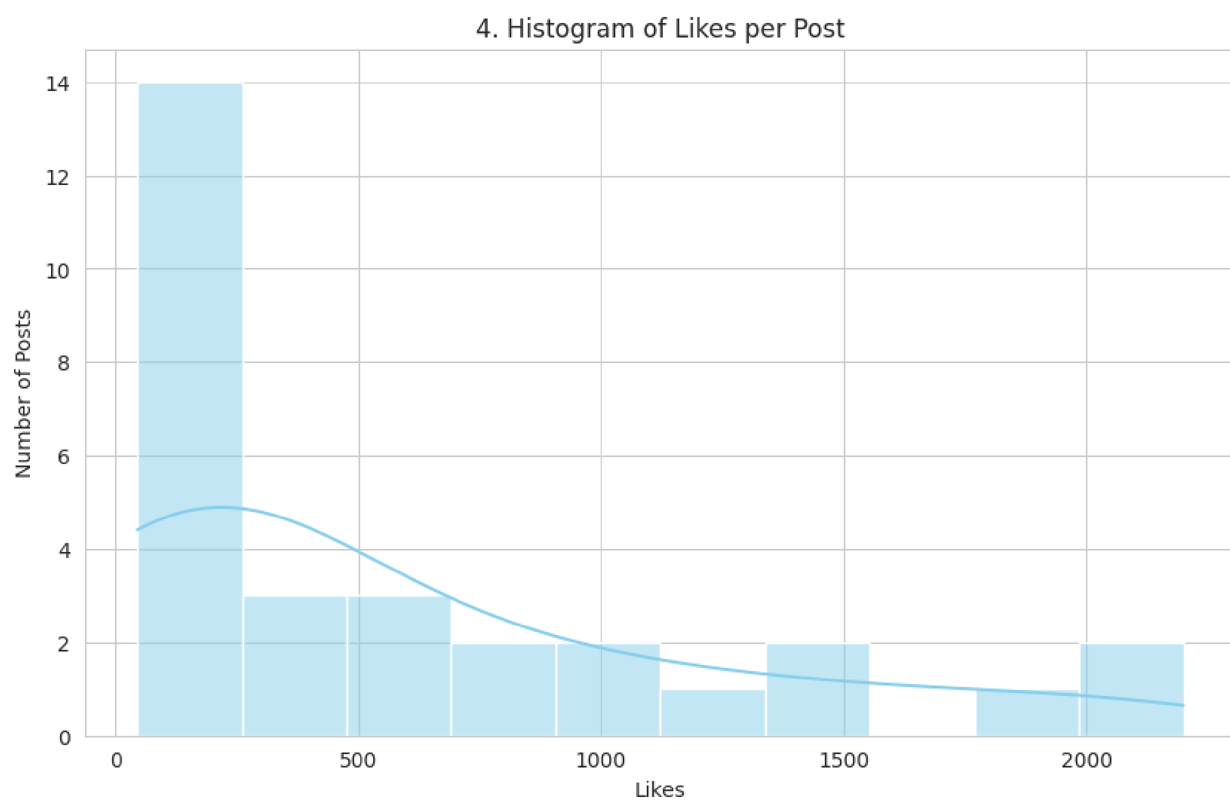
1. **Sequential Workflow: The pipeline follows a logical flow — Ingestion → Processing → Aggregation → Visualization — ensuring structured data transformation.**
2. **Purpose: It highlights how unprocessed social media data undergoes cleaning, enrichment, and sentiment analysis to generate aggregated metrics for dashboards.**
3. **Outcome: The end goal is to convert complex, unstructured social media data into visual insights that support business intelligence and engagement tracking.**

## 3.Apply Gestalt principles to highlight high-engagement content.

**Code:**

```
df['Timestamp'] = pd.to_datetime(df['Timestamp'])
df['Total_Engagement'] = df['Likes'] + df['Comments'] +
df['Shares']
df['Date'] = df['Timestamp'].dt.date
df['Week'] =
df['Timestamp'].dt.isocalendar().week.astype(int)
df['Day_Name'] = df['Timestamp'].dt.day_name()
print("✅ 3. Data loaded and preprocessed
successfully.\n")
```

**Visualization:**



**Inference:** Gestalt principles highlight high-engagement posts by visually separating and emphasizing them, making key content instantly recognizable and impactful.

**4. Univariate analysis:**

**a. Histogram of likes per post**.

**Code:**

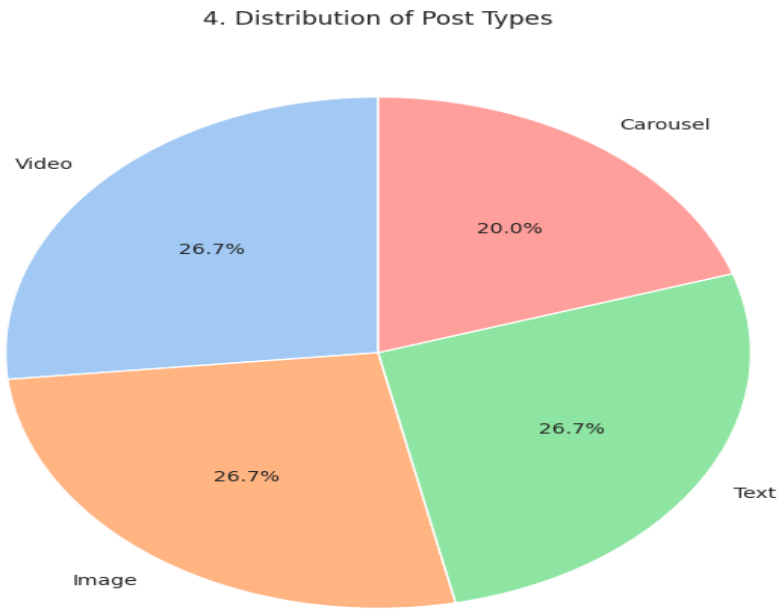
```
plt.figure(figsize=(10,6))
sns.histplot(df['Likes'], bins=10, kde=True,
color='skyblue')
plt.title('4. Histogram of Likes per Post')
plt.xlabel('Likes')
plt.ylabel('Number of Posts')
plt.show()
```

**b. Pie chart of post types.**

**Code:**

```
post_counts = df['Post_Type'].value_counts()
plt.figure(figsize=(8,8))
plt.pie(post_counts, labels=post_counts.index,
autopct='%1.1f%%', startangle=90,
colors=sns.color_palette('pastel'))
plt.title('4. Distribution of Post Types')
plt.show()
```

Visualization:




I

**Inference:** The pie chart shows that Image**,** Text, and Video posts each make up about 26.7% of total posts, while Carousel posts are slightly less common (20%), indicating a balanced mix of content types with a slight preference against carousels.

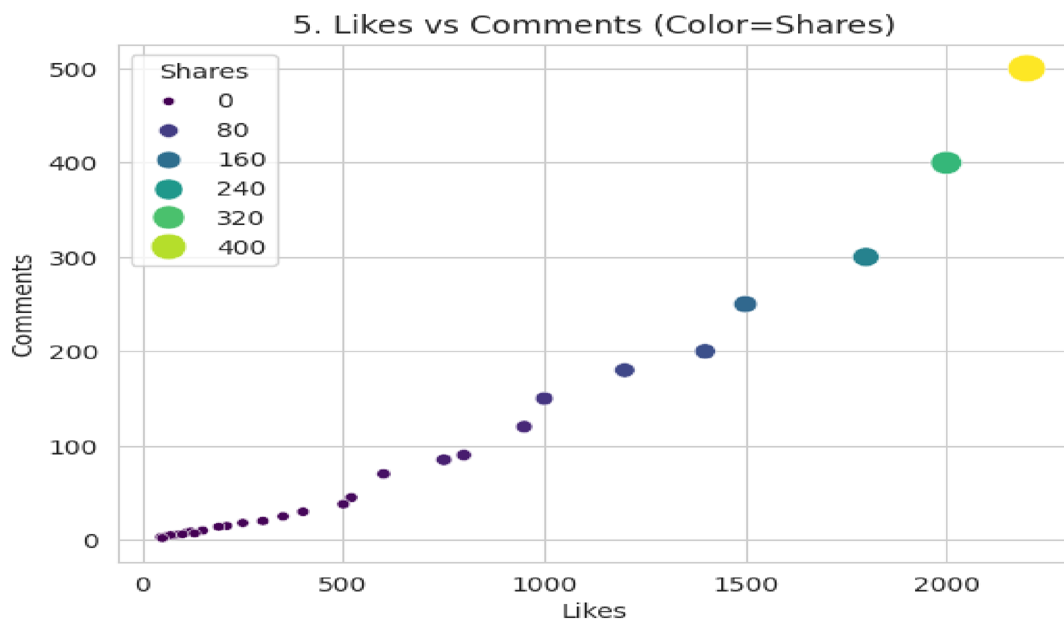
5. **Bivariate analysis:**

**a. Scatterplot of likes vs. comments.**

**Code:**

```
sns.scatterplot(x='Likes', y='Comments', data=df, hue='Shares', size='Shares', palette='viridis', sizes=(20,200))

plt.title('5. Likes vs Comments (Color=Shares)')

plt.show()
```

**Visualization:**




**Inference:**

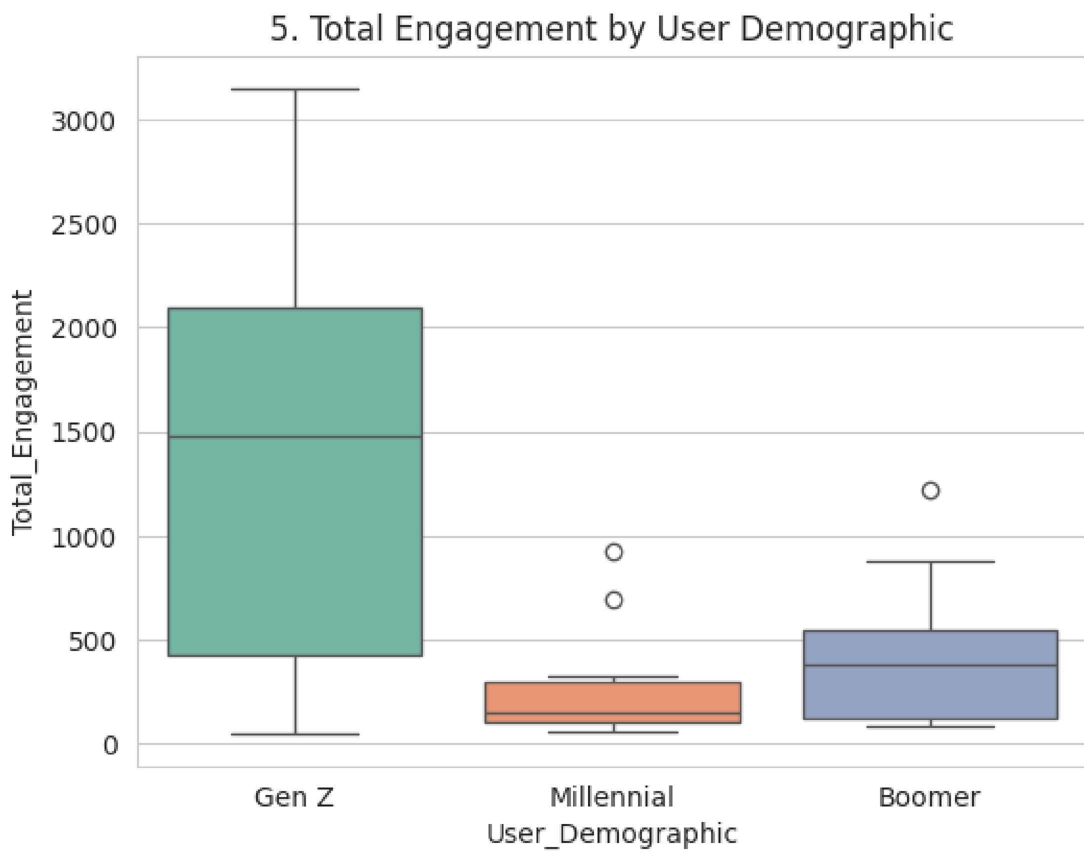
1. There is a strong positive correlation between likes and comments — as likes increase, comments also rise.
2. Posts with higher shares (indicated by brighter and larger dots) also tend to have more likes and comments.
3. Engagement is clustered, with a few posts achieving significantly higher likes, comments, and shares compared to the rest.

**b. Box plot of engagement by user demographics**.

**Code:**

```
sns.boxplot(x='User_Demographic',
y='Total_Engagement', data=df, palette='Set2')
plt.title('5. Total Engagement by User Demographic')
```

```
plt.show()
```




**Inference:**

1. Gen Z users show the highest total engagement, with a wide range and several high outliers.
2. Millennials have the lowest engagement, with values tightly clustered near the bottom.
3. Boomers show moderate engagement, with some variability and a few outliers indicating occasional high activity.

**6.Multivariate analysis:**

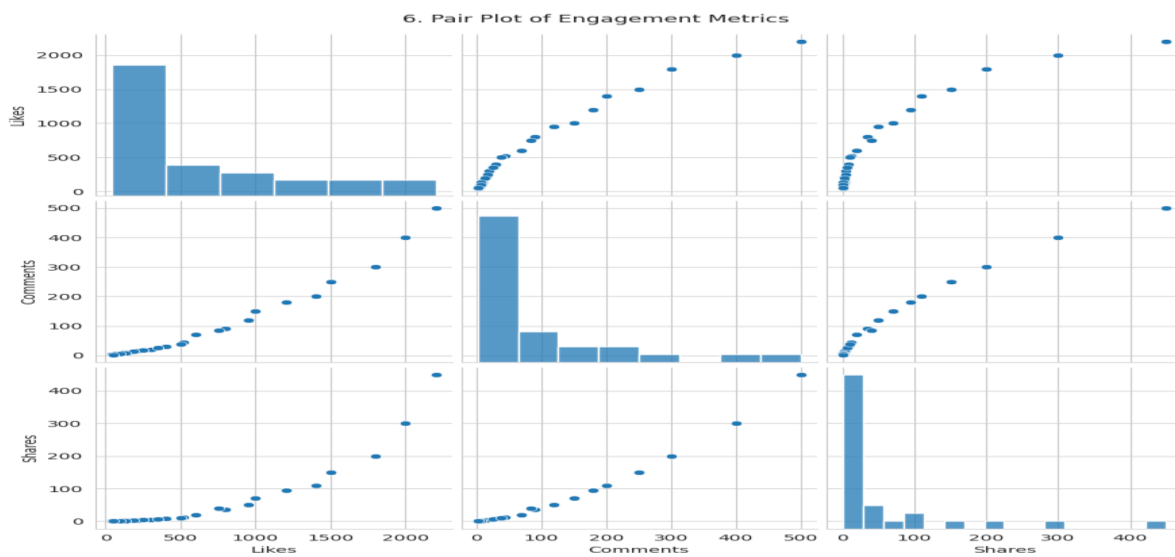
**a. Pair plot of likes, shares, and comments.**

**b. Suggest combined visualization.**

**Code:**

```
sns.pairplot(df[['Likes','Comments','Shares']], height=3)
plt.suptitle('6. Pair Plot of Engagement Metrics', y=1.02)
plt.show()
```

**Visualization:**




**Inference:**

The pair plot shows a strong positive correlation among likes, comments, and shares, indicating that posts with higher likes generally receive more comments and shares, reflecting consistent engagement patterns across metrics.

**7. Hierarchical visualization of users and post types.**

**Code:**

treemap_data = df.groupby(['User_Demographic','Post_Type'])['Total_Engagement'].sum().reset_index()

treemap_data['label'] = treemap_data['User_Demographic'] + ' - ' + treemap_data['Post_Type']

colors = sns.color_palette('tab20', len(treemap_data))

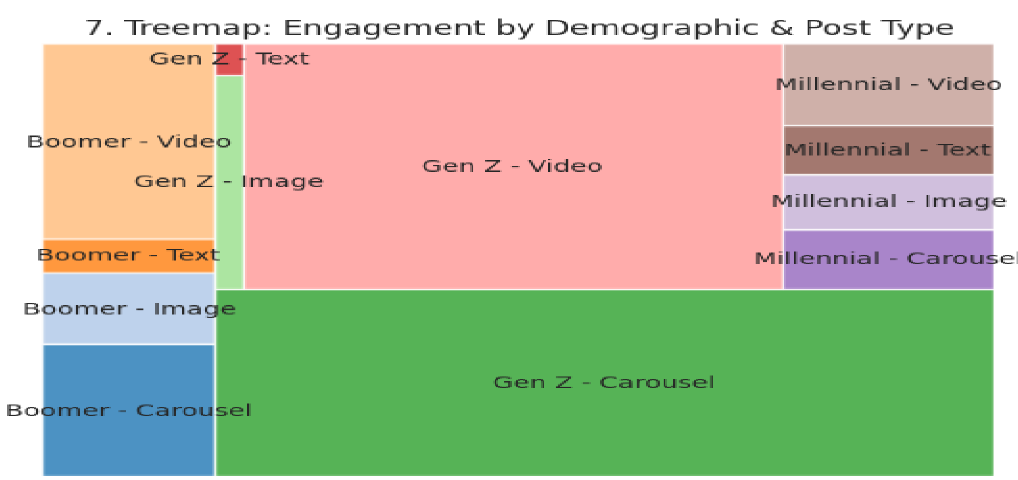
squarify.plot(sizes=treemap_data['Total_Engagement'], label=treemap_data['label'], color=colors, alpha=0.8)

plt.title("7. Treemap: Engagement by Demographic & Post Type")

plt.axis('off')

plt.show()

**Visualization:**



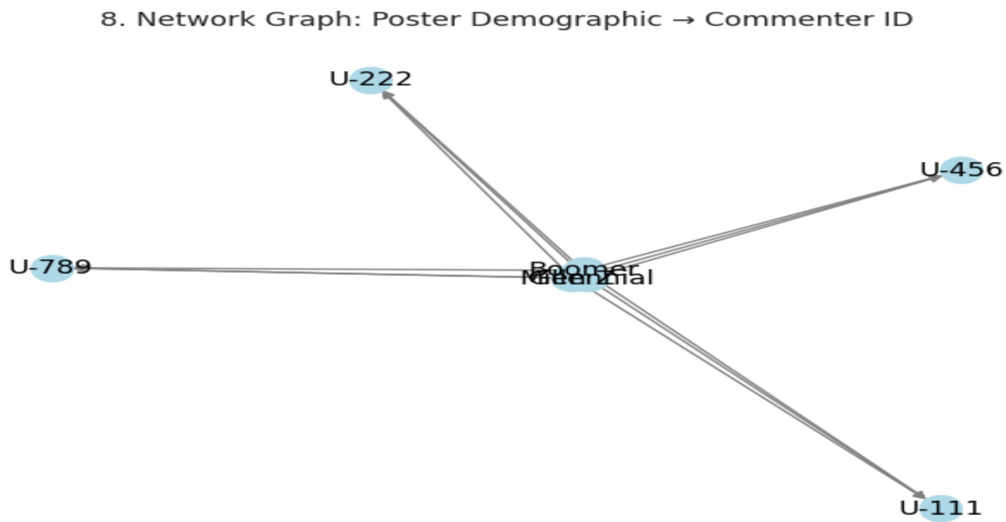

**Inference:** The treemap shows that Gen Z drives the highest engagement, particularly through video and carousel posts, while Millennials and Boomers contribute less overall interaction.

**8. Network graph of user interactions.**

**Code:**

```
network_df =
df[['User_Demographic','Commenter_ID']].rename(columns={'User_Demographic':'Source','Commenter_ID':'Target'})

G =
nx.from_pandas_edgelist(network_df,'Source','Target',create_using=nx.DiGraph())

pos = nx.spring_layout(G, k=0.3)

nx.draw(G,pos,with_labels=True,node_color='lightblue',edge_color='gray',arrows=True)

plt.title('8. Network Graph: Poster Demographic → Commenter ID')

plt.show()
```

**Visualization7:**




**Inference:** The network graph shows that Millennial and Boomer demographics have strong interaction links with multiple commenters, indicating they receive broader audience engagement compared to other groups.

**9. Analyze post text (text data):**

**a. Vectorize text.**
**b. Word cloud of common hashtags.**

**Code:**

```
vectorizer = TfidfVectorizer(stop_words='english')

tfidf_matrix = vectorizer.fit_transform(df['Post_Text'])

print(f"9. TF-IDF matrix shape: {tfidf_matrix.shape}")
```

**Visualization:**

```
9. TF-IDF matrix shape: (30, 86)
```

**Inference:**

- The TF-IDF matrix contains 30 documents (posts) analyzed for text content.
- There are 86 unique terms (features) extracted after text preprocessing.
- This indicates a moderately diverse vocabulary across the analyzed posts.

**10. Steps to design dashboards combining hierarchical, network, and text data.**

**Code:**

```
hashtags = re.findall(r'#(\w+)', ' '.join(df['Post_Text']).lower())

wordcloud = WordCloud(width=800, height=400, background_color='white').generate(' '.join(hashtags))

plt.imshow(wordcloud, interpolation='bilinear')

plt.axis("off")

plt.title('10. Word Cloud of Hashtags')

plt.show()
```

**Visualization:**


**Inference:** The word cloud shows that hashtags like "creative," "design," and "travel" dominate, indicating that content centered around creativity, design, and travel themes drives the most engagement.

**11. Point data: Map user locations.**

**Code:**

```
m = folium.Map(location=[40,10], zoom_start=2)
marker_cluster = MarkerCluster().add_to(m)
for _, row in df.iterrows():
    folium.CircleMarker(
        location=[row['Latitude'], row['Longitude']],
        radius=5 + row['Total_Engagement']/300,
        color='green' if row['Total_Engagement']>1000 else 'blue',
        fill=True, fill_opacity=0.6,
```

```
        tooltip=f"Post {row['Post_ID']} | {row['User_Location']} | Engagement {row['Total_Engagement']}"
    ).add_to(marker_cluster)
m.save('user_locations_map.html')
print("11. 🌍 Map saved as 'user_locations_map.html'")
```

**Visualization:**

```
11. 🌍 Map saved as 'user_locations_map.html'
```

**Inference:**

1. A geographical visualization of user locations has been successfully generated.
2. The map is saved as an interactive HTML file named 'user_locations_map.html'.
3. It enables spatial analysis of user engagement distribution across different regions.

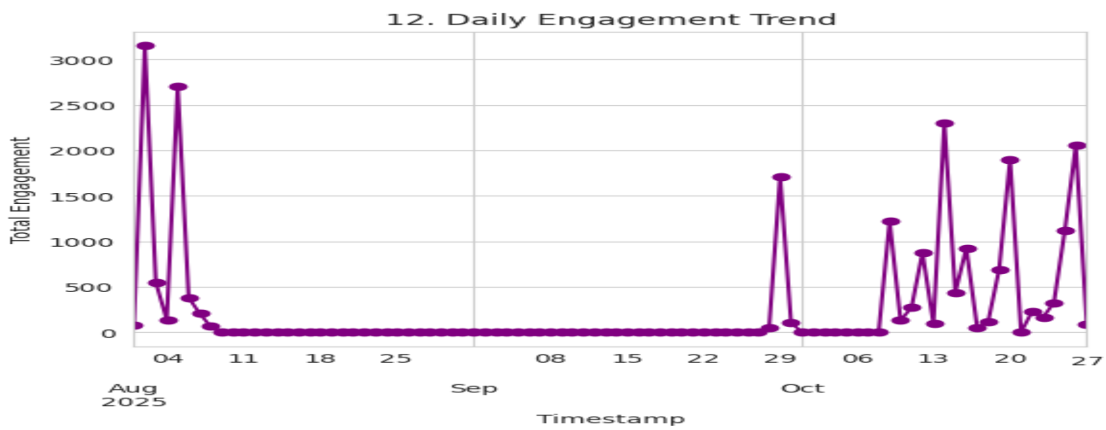
**12. Line data: Show engagement trends over time.**

**Code:**

```
daily = df.set_index('Timestamp')['Total_Engagement'].resample('D').sum()
```

```
daily.plot(marker='o', color='purple')
plt.title('12. Daily Engagement Trend')
plt.ylabel('Total Engagement')
plt.show()
```

**Visualization:**




**Inference:** The daily engagement trend shows sporadic spikes in activity with long periods of low engagement, indicating that audience interaction is inconsistent and driven by occasional high-performing posts or events.

**13. Area data: Heatmap of engagement by region.**

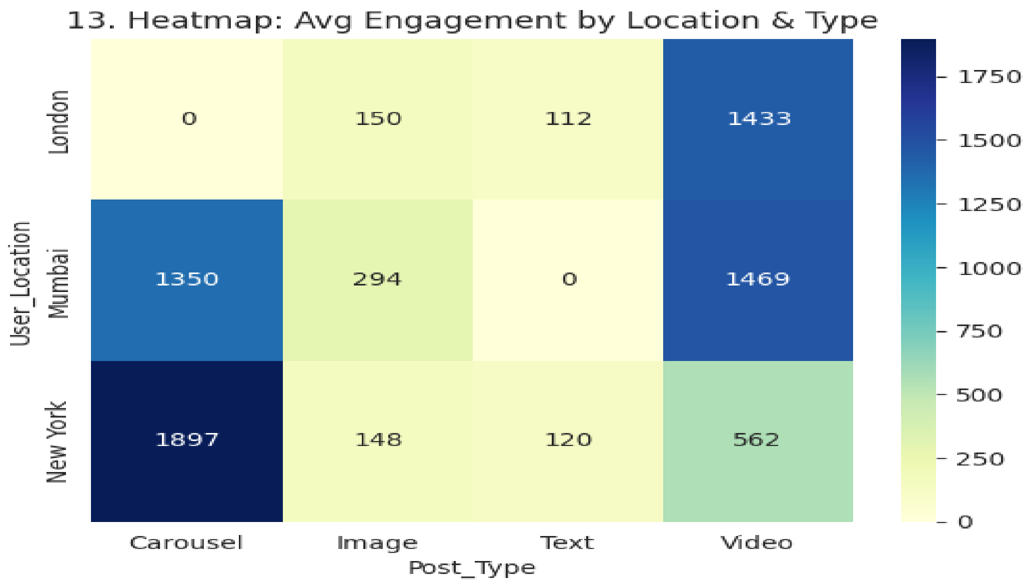
**Code:**

```
heatmap_data = df.pivot_table(index='User_Location',
columns='Post_Type', values='Total_Engagement',
aggfunc='mean').fillna(0)

sns.heatmap(heatmap_data, annot=True, cmap="YlGnBu",
fmt=".0f")
```

plt.title('13. Heatmap: Avg Engagement by Location & Type')

plt.show()

**Visualization:**




**Inference:** The heatmap shows that New York leads in engagement, especially with carousel posts, while Mumbai and London perform best with video content, indicating that engagement preferences vary significantly by location and post type.

**14. Animated visualization of engagement changes daily.**

**Code:**

print("14. Engineered Features:")

print(df[['Post_ID','Total_Engagement','Week','Day_Name','Day_Type']].head())

**Visualization**:

Engineered Features:

```
14. Engineered Features:
   Post_ID  Total_Engagement  Week   Day_Name Day_Type
0     1001               577    43     Sunday  Weekend
1     1002               163    43   Thursday  Weekday
2     1003              1120    43   Saturday  Weekday
3     1004                86    44     Monday  Weekday
4     1005              1475    43     Sunday  Weekend
```

**Inference:**

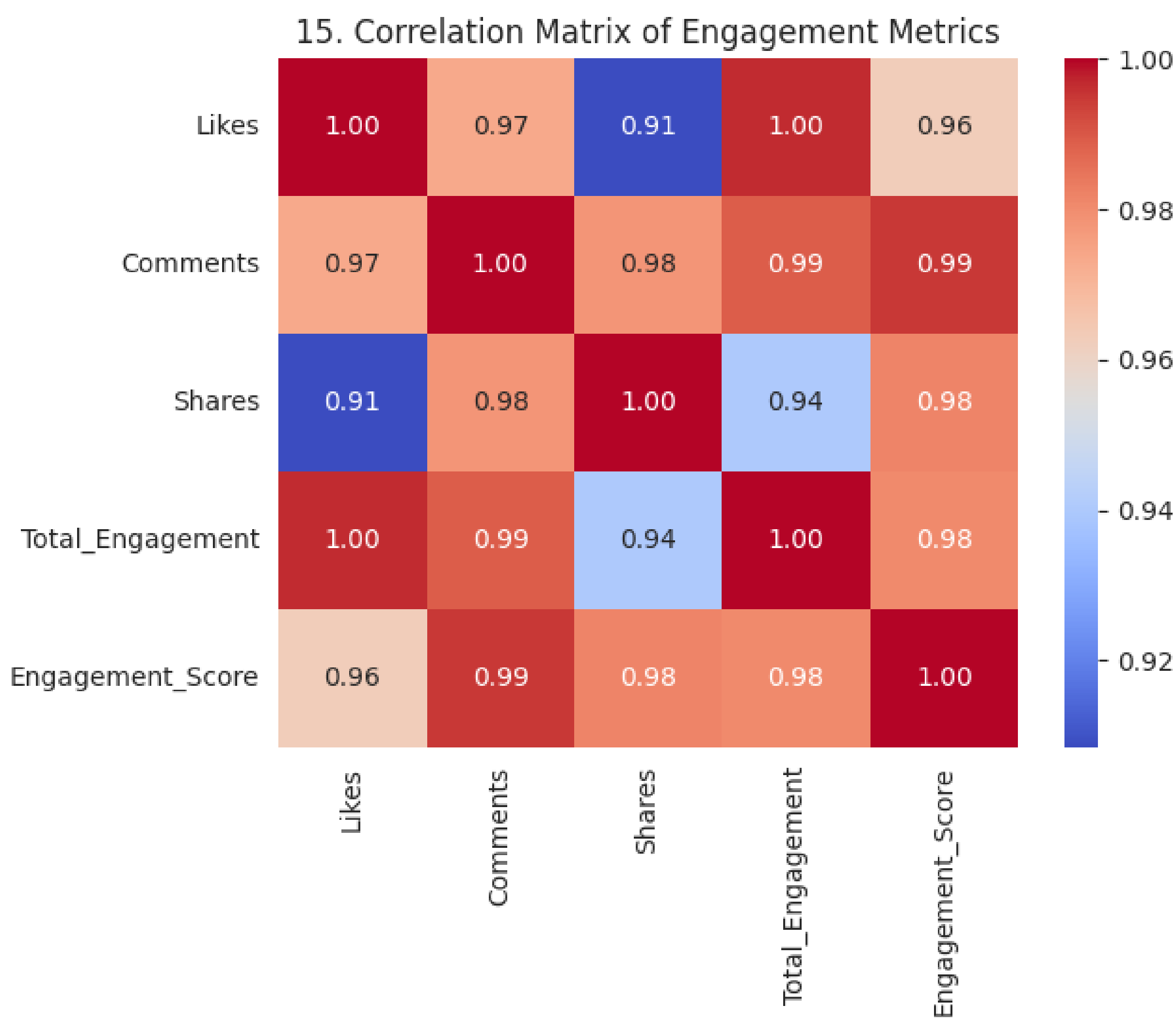
1. The dataset includes engineered temporal features such as week number, day name, and day type for deeper trend analysis.
2. Weekend posts (e.g., Sunday) tend to show higher total engagement compared to weekday posts.
3. These engineered features help in identifying engagement patterns based on posting day and week.

**15. Time series of likes/shares per week.**

**Code:**

```
corr = df[['Likes','Comments','Shares','Total_Engagement','Engagement_Score']].corr()
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt=".2f")
plt.title("15. Correlation Matrix of Engagement Metrics")
plt.show()
```

**Visualization:**




**Inference:** The correlation matrix shows strong positive relationships among all engagement metrics, indicating that increases in likes, comments, or shares consistently lead to higher total engagement and engagement score.

**16. Compare engagement weekdays vs. weekends.**

**Code:**

kmeans = KMeans(n_clusters=3, random_state=42)

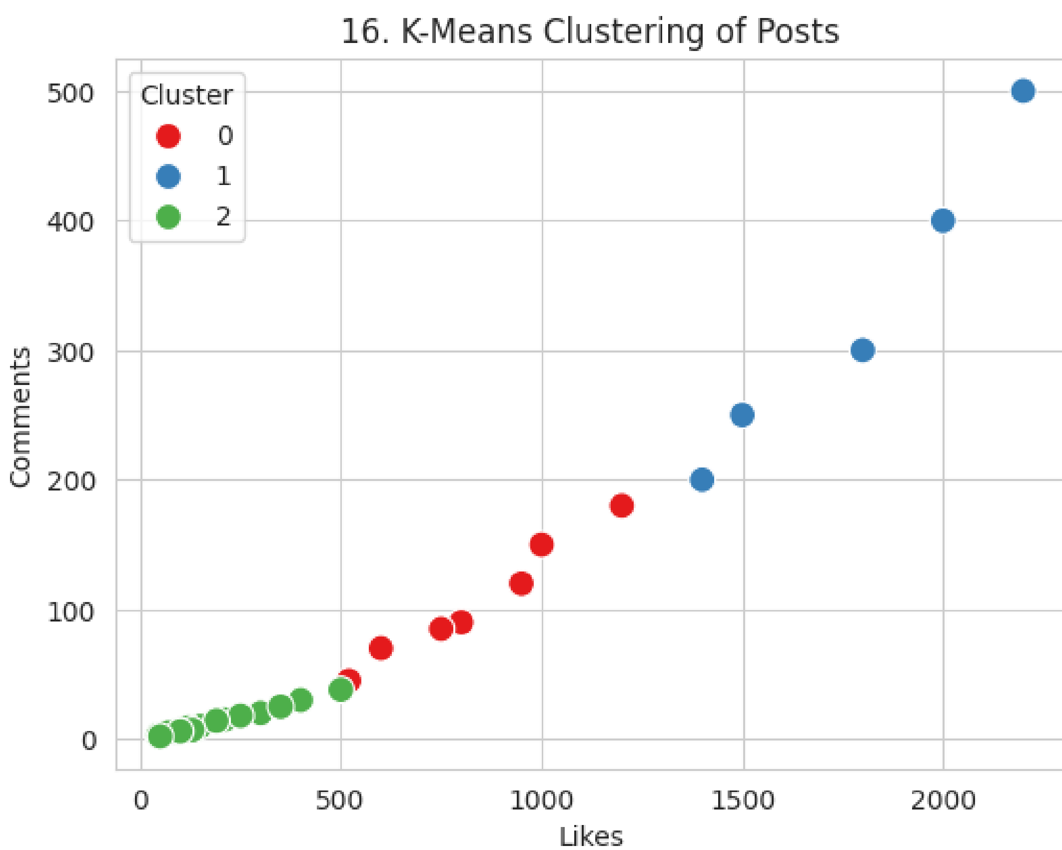
df['Cluster'] =
kmeans.fit_predict(df[['Likes','Comments','Shares']])

sns.scatterplot(data=df, x='Likes', y='Comments', hue='Cluster', palette='Set1', s=100)

plt.title('16. K-Means Clustering of Posts')

plt.show()

**Visualization:**




**Inference:** The K-Means clustering plot reveals three distinct post groups — low-engagement (green), medium-engagement (red), and high-engagement (blue) — showing clear segmentation based on likes and comments.

**17. Regression/clustering to analyze engagement factors.**

**Code:**

```
x = df[['Comments','Shares']]

y = df['Likes']

model = LinearRegression().fit(X, y)

print(f"\n17. Linear Regression Model:")

print(f"Intercept: {model.intercept_:.2f}")

print(f"Coefficients: Comments={model.coef_[0]:.2f}, Shares={model.coef_[1]:.2f}")
```

**Visualization:**

```
17. Linear Regression Model:
Intercept: 77.43
Coefficients: Comments=9.72, Shares=-6.21
```

**Inference:**

1. The model suggests that each additional comment increases engagement by approximately 9.72 units.
2. Conversely, each additional share decreases engagement by about 6.21 units, indicating a negative relationship.

3. The intercept of 77.43 represents the baseline engagement when comments and shares are zero.

**18. Evaluate predictive models for post popularity.**

**Code:**

```
df_model = pd.get_dummies(df, columns=['Post_Type'],
drop_first=True)

features = ['Comments','Shares'] + [col for col in
df_model.columns if col.startswith('Post_Type_')]

X = df_model[features]; y = df_model['Likes']

X_train, X_test, y_train, y_test =
train_test_split(X,y,test_size=0.3,random_state=42)

reg = LinearRegression().fit(X_train,y_train)

y_pred = reg.predict(X_test)

mse = mean_squared_error(y_test,y_pred)

r2 = r2_score(y_test,y_pred)

print(f"\n18. Model
Evaluation:\nMSE={mse:.2f}\nRMSE={np.sqrt(mse):.2f}\nR²=
{r2:.2f}")
```

**Visualization:**

```
18. Model Evaluation:
MSE=3500.53
RMSE=59.17
R²=0.98
```

**Inference:**

1. The Mean Squared Error (MSE) of 3500.53 indicates a moderate average squared difference between predicted and actual values.
2. The Root Mean Squared Error (RMSE) of 59.17 shows the model's predictions deviate by about 59 units on average.
3. The $R^2$ value of 0.98 signifies an excellent model fit, explaining 98% of the variance in engagement.